HEALPix Facility Installation Guidelines
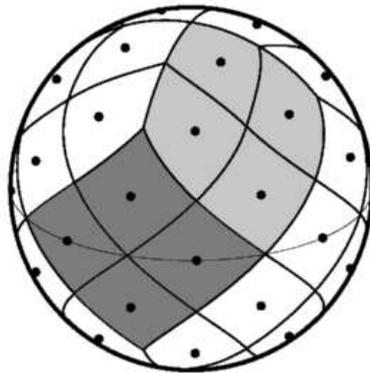
Revision:	Version 2.15a; June 18, 2010
Prepared by:	Eric Hivon, Anthony J. Banday, Matthias Bartelmann, Benjamin D. Wandelt, Frode K. Hansen and Krzysztof M. Górski
Abstract:	This document describes the installation for the **HEALPix** facilities.

This document describes the installation for the **HEALPix** facilities.

TABLE OF CONTENTS

1 Introduction

In this document the installation procedure for the **HEALPix** distribution is outlined. **HEALPix** comprises a suite of Fortran 90, C++, IDL and Java routines providing both stand-alone facilities and callable subroutines as an alternative for those users who wish to build their own tools. A set of C subroutines and functions is also provided.

The distribution can be downloaded as a gzipped and tarred file, which can be unpacked by executing the commands
```
% gunzip Healpix_2.15a.tar.gz
% tar -xpf Healpix_2.15a.tar
```
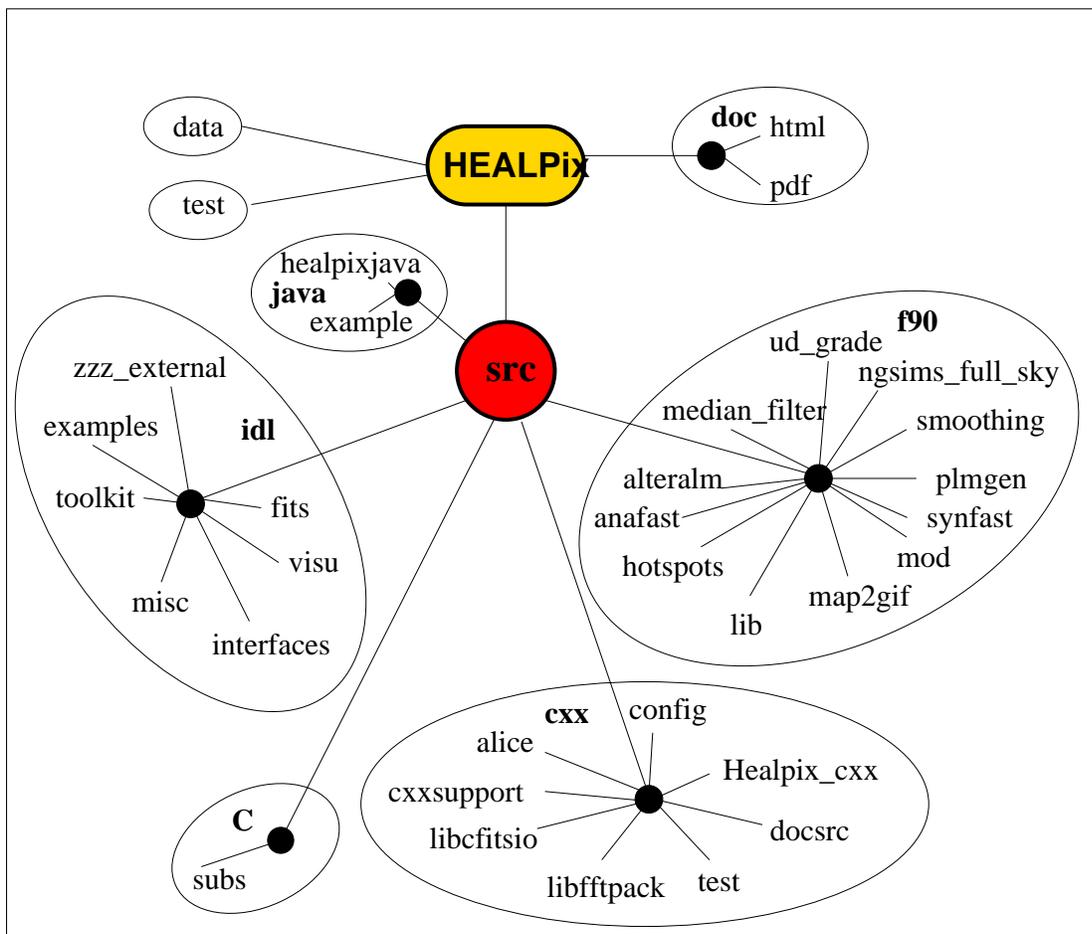

Figure 1: The directory structure for the **HEALPix** distribution.

The unpacked distribution has a directory structure as shown in Figure 1.

As with most freely available software, the distribution comes with caveats, the major one being that although we have attempted to automate the installation as much as possible, not all eventualities can ever be foreseen. We have tested the installation on the following platforms:

AIX, IRIX, IRIX64, Linux, SunOS, ALPHA and Darwin (MacOS)

There may be problems in the facility build due to the local system configuration which is beyond our control.

2   Installation Requirements

The major part of the **HEALPix** distribution is written in both **Fortran 90** and **C++** and so the appropriate compiler(s) must be present (Linux and Darwin users should look at Section 8.3.4 about free F90 compilers. Microsoft Windows users should look at Section 6.1). Many visualisation tools and map manipulation routines are provided in **IDL** (please note that at least version 6.1 is required) and **Java**. Some of the **HEALPix** routines are also available in **C**.

*This section and the next focus on the compilation and installation of the **C**, **C++**, **Fortran 90** and **IDL** routines. For more information on the **Java** routines see table 1*

Healpix Package	Information on installation	Information on routines
Fortran 90	This document	"Fortran Facilities" and "Fortran Subroutines" documents
IDL	This document	"IDL Facilities"
C++	This document, or `src/cxx/README.compilation`	"C++ Facilities and Subroutines" (HTML only)
java	`src/java/README`	"Java Overview" (HTML only)
C	This document, or `src/C/README`	"C Subroutines Overview"

Table 1: Documentation on the installation and usage of the different packages

The configure script is written in the Bourne shell. The script attempts to generate a `Makefile` which is tailored to one of the above Operating Systems (OS's) and using `Makefile.in` as a template for non system-specific statements. Only the basic UNIX

make facility is required to build the software, although we do still recommend the GNU make facility (ftp://ftp.gnu.org/gnu/make/). In addition, several environment configuration files and an IDL startup file are generated. These automatically establish various environment variables and aliases to make the use of the **HEALPix** package simpler.

The **HEALPix Fortran 90**, **C++** and **C** distributions also require the publicly available CFTISIO library:

Software Package	Source
CFITSIO V 2.xx or 3.xx library	http://heasarc.gsfc.nasa.gov/docs/software/fitsio/

The **IDL** visualization software is commercially available at

Software Package	Source
IDL V 6.1 or more	http://www.ittvis.com/

while the GNU Data Language **GDL**, a *free* clone of IDL 6.0, can also be used (with some caveats, see §6.8) and can be downloaded for free from

Software Package	Source
GDL 0.9rc3 or more	http://sourceforge.net/projects/gnudatalanguage/

As it was already the case in version 1.20, users no longer need to acquire the IDL Astronomy User's Library (http://idlastro.gsfc.nasa.gov/homepage.html) or the COBE (IDL) Analysis Software (http://lambda.gsfc.nasa.gov/product/cobe/cgis.cfm), although we do recommend these packages to the user. The 100-odd routines required for version 2.15a are contained in the subdirectory Healpix_2.15a/src/idl/zzz_external. These procedures are included in the **HEALPix** package unchanged and solely for the purpose of making it self contained. In this way, we remove the burden of installation of additional libraries from the end user.

A parallel implementation (based on OpenMP, for shared memory architectures) of the Spherical Harmonics Transforms involved in synfast, anafast, smoothing, plmgen and alteralm is now available by default and can be readily compiled and used with the standard installation script.

A set of routines with MPI parallelization (for distributed memory architectures) is also available for Spherical Harmonics Transform, thanks to the work of H.K. Eriksen (h.k.k.eriksen@astro.uio.no) and Snorre Boasson (ITEA, NTNU). See the F90 subroutines

documentation for more information on how to use those routines in your code.

We found that it was remarkably difficult to find random number generators in the public domain which are simple yet powerful and easy to use. We are providing one (both in C++ and F90) which is an adaptation of an xorshift generator described in Marsaglia (Journal of Statistical Software 2003, vol 8). It has a theoretical period of $2^{128} - 1 \approx 3.4 \ 10^{38}$.

3   The Installation Procedure

If the user has one of the supported OS's, then installation proceeds utilizing the following commands. If your OS is not supported, the configuration step should be omitted, `Makefile.in` should be copied as `Makefile` and explicitly tailored to the user environment.

`% ./configure [-L]`	uses `Makefile.in` as a template to build the correct Makefile (from user inputs as required), it will also configure the IDL routines
`% make`	builds all the facilities
`% make test`	tests all the facility previously compiled
`% make clean`	removes object files
`% make tidy`	removes object files, executables and libraries
`% make distclean`	same as above and restores the directories to the state of the original distribution

These different steps are detailled below.

3.1   ./configure [-L]

The `./configure` script manages the configuration of the C, C++, Fortran90 and IDL suites of routines and facilities.

Since v2.11, it accepts the `-L` option to write the **HEALPix** specific configuration files into the **HEALPix** directory itself rather than in installer's home directory (see § 3.1.1). Using the `-L` option is recommended when doing a *project* or *system wide* installation of **HEALPix** to be accessed by several different users.

An online help is available with `./configure -h`.

3.1.1   Configuration profile

A feature introduced in previous releases and enhanced since v2.10, is that the configure script creates a shell configuration file
(located in `${HOME}/.healpix/2_12_⟨OS_TYPE⟩/config` or in
`${HEALPIX}/confdir/2_12_⟨OS_TYPE⟩/config` if `./configure -L` was used) according

to shell type in which various environment variables and aliases are defined for your convenience. If you agree upon prompting, it will also change your default system profile during installation to automatically source this profile. If you do not agree to this change, you will need to explicitly source the configuration file above for any session in which you intend to run **HEALPix** facilities. **In particular, you will have to make sure that the** `HEALPIX` **system variable is correctly defined (as the full path to the HEALPix directory) before running the package**.

3.1.2   C configuration

The `./configure` script will ask for the C compiler and options to be used, and for the full path of an installed `cfitsio` library to link to. By default, only a static library is created, but the user can also ask for a shared (Unix/Linux systems) or dynamic (Darwin) library. After compilation (see `make` section) and linking, all libraries will be in `${HEALPIX}/lib/chealpix.*` .

3.1.3   C++ configuration

The `./configure` script will ask for the full path to a **cftisio tar ball**, and then provide a choice a predefined targets corresponding to different combinations of C++ compilers and options. Each of those targets is defined in a configuration file located in `Healpix_2.15a/src/cxx/config/config.`*target*. The user can therefore add new targets or edit existing ones, and the `./configure` script will update its menu accordingly. If a fairly recent version (3.2 or higher) of gcc and g++ is installed on the system, the target "generic_gcc" should always work. The environment variable HEALPIX_TARGET will be set to the value of the chosen target.
If the HEALPIX configuration file is sourced as described in § 3.1.1, the full path to the C++ executables will be added to the environment PATH variable.

3.1.4   Fortran 90 configuration

When you run `./configure` on a supported system you will be prompted to enter compiler optimisation flags. We have not attempted to provide the best optimisation flags for all operating systems. The configure script will have a guess at optimisation options for some systems, but it is up to the user to figure out an optimal set[1]. From our experience, we have not found significant accumulation of numerical error even when using the most aggressive optimisation level available.

---

[1]In particular, the Intel Fortran Compiler, available for free for PC's with Intel-like processors, have a set of optimization options for each of Intel processor families (Pro, II, MMX, 4). Please consult the online help (`ifort -help`) or PDF documentation (`/opt/intel_fc_80/doc/` or `/opt/intel/fc/9.*/doc`) or HTML documentation (`/opt/intel/fce/10.*/doc/Doc_Index.htm`) for further information.

If the HEALPIX configuration file is sourced as described in § 3.1.1, the full path to the F90 executables will be added to the environment PATH variable.

3.1.5   IDL configuration

You will be asked for the external applications you want to use to visualize the Postscript and PNG files created by IDL.
If the HEALPix configuration file is sourced as described in § 3.1.1, the aliases `hidl` and `hidle` are also defined to give you access to HEALPIX routines from IDL.

3.2   Compilation and installation

The
```
make
```
command will compile one or several of the C, C++ and F90 packages depending on what was configured with the `./configure` script. Specific packages can be compiled with the respective commands

```
make c-all
make cpp-all
make f90-all.
```

To perform several compilation jobs simultaneously, the command `make -j [jobs]` can be used.

Please neglect any possible warnings at compile time. If you run into trouble please refer to the section **Troubleshooting and further information**.

After running `make`, the user must re-login to ensure that the new profiles built by the installation procedure are correctly sourced. Only then will the user have full access to the specific **HEALPix** environment variables etc.

3.3   Testing the installation

All installed libraries and executables can be tested with

```
make test
```

while specific tests of the C, C++ and Fortran products can be performed with, respectively

```
make c-test
make cpp-test
make f90-test
```

code & parameter file	output data	reference data	output image	reference image
synfast syn.par	test_map.fits	map.fits	test_map.gif	map.gif
	test_alm.fits	alm.fits	*NA*	*NA*
smoothing smo.par	test_sm.fits	map_sm.fits	test_sm.gif	map_sm.gif
ud_grade udg.par	test_LOres.fits	map_LOres.fits	test_LOres.gif	map_LOres.gif
hotspot hot.par	test_ext.fits	map_ext.fits	test_ext.gif	map_ext.gif
	test_max.asc	max.asc	*NA*	*NA*
	test_min.asc	min.asc	*NA*	*NA*
anafast ana.par	test_cl.fits	cl_out.fits	*NA*	*NA*
alteralm alt.par	test_almdec.fits	almdec.fits	*NA*	*NA*
median_filter med.par	test_mf.fits	map_mf.fits	test_mf.gif	map_mf.gif
sky_ng_sim ngfs.par	test_ngfs.fits	map_ngfs.fits	test_ngfs.gif	map_ngfs.gif

Table 2:   Data files and images produced by the Fortran codes during the tests, and the respective reference files to which they can be compared. All the files listed are located or produced in the `Healpix_2.15a/test` directory. The GIF images of full sky maps were produced using `map2gif`. *NA*: No image available, because the data set is not a sky map

For the latter, Table 2 lists the codes tested with the parameter files used, as well as the data files produced and the respective reference files.

Notes:

- the input power spectrum (in `Healpix_2.15a/test/cl.fits`) used to generate the Fortran90 test maps is currently the WMAP 1yr best fit, in $(\mu K)^2$, and is therefore different from the one included in releases 1.* (that can still be found in `cl_old.fits`). See http://lambda.gsfc.nasa.gov/ for details on WMAP and its data products.

- the file `Healpix_2.15a/test/wmap_lcdm_sz_lens_wmap5_cl_v3.fits` was added for convenience, even though it is currently *NOT* used for any of the simulated test maps.

  It has been adapted to run with **HEALPix** from WMAP 5yr best fit model for Λ-CDM + SZ + lensing with B mode = 0, in $(\mu K)^2$ (input file: http://lambda.gsfc.nasa.gov/data/map/dr3/dcp/params/c_l/wmap_lcdm_sz_lens_wmap5_-cl_v3.dat). For the value of the cosmological parameters, see http://lambda.gsfc.nasa.gov/product/map/dr3/params/lcdm_sz_lens_wmap5.cfm

In order to test the new **HEALPix** profile set-up one can then attempt to run any C++ or F90 facility from any directory on your system. Similarly, IDL should be tested by invoking `hidl` or `hidlde`.

3.4   Cleaning up

Three levels of cleaning are available:

```
make clean
```

will remove the intermediate files created during compilation, such as object files, (Fortran) modules files, ... found in the source or build directories;

```
make tidy
```

same as above, and will also remove the **HEALPix** executables, libraries and module and/or include files;

```
make distclean
```

will return the **HEALPix** directory to its original 'distribution' state by discarding the same files as above, as well as the executable and library directories and the top level Makefile.

4   Upgrading from 2.0 to 2.1

The internal structure of release 2.1 is quite different from release 2.0 and to avoid confusion during the compilation we highly recommend to put the new release in a *different* directory, rather than putting the new package on top of the old one. If you actually change the name of the 'active' **HEALPix** directory care must be taken that all references to the old directory are removed from your system profile before adding the new ones (see Note on *Re*-installation).

5   A Note on *Re*-installation

As a result of the line added to your shell profile which explicitly sources the **HEALPix** profile, care must be taken if the package is reinstalled in a different directory. If such reinstallation is desired, the included line must be removed from your system profile, allowing the corrected version to be added.

6   Troubleshooting and further information

This section contains a list of difficulties which we have dealt with. It is by no means exhaustive. A troubleshooting forum has been established at http://healpix.jpl.nasa.gov/healpixSoftwareProblemsBugsSolutions.shtml, where we list current questions and solutions to known problems (for a given release). A bug tracker is also available on SourceForge at this address. If the problem you encounter is not addressed in those forums nor below, please contact *healpix* at *jpl.nasa.gov*

6.1   Installation under Microsoft Windows

The installation and usage of HEALPix require many standard Unix/Linux tools (such as `sh, make, awk, grep, sed, ls, wc, cat, more, nm, ar`) as well as C, C++ and Fortran compilers. To install it under Windows, you will need to

- Install Cygwin on your machine (see http://cygwin.com/). In addition to the default packages, you need at least the `binutils, coreutils, util-linux, bash, gawk, grep, make` and `sed` packages, as well as `gcc` and gcc-g++ packages, all available at http://cygwin.com/packages/.

- Install   the   latest   gfortran   binaries   for   Cygwin   from http://quatramaran.ens.fr/coudert/gfortran/, following the instructions at http://gcc.gnu.org/wiki/GFortranBinaries.

- Unpack the HEALPix software package

- Run configure as you would on other platforms

- The C++ code can be compiled using HEALPIX_TARGET=generic_gcc

6.2   Problems with CFITSIO

6.2.1   Compilation of CFITSIO Fortran wrappers

The most common problem with the Fortran **HEALPix** compilation will produce messages like:

```
ld: Undefined symbols:
 _ftbnfm_
 _ftclos_
 _ftcrhd_
 _ftdkey_
 ...
```

or

```
fitstools.f90: undefined reference to 'ftdkey_'
fitstools.f90: undefined reference to 'ftbnfm_'
fitstools.f90: undefined reference to 'ftclos_'
 ...
```

or

```
Undefined symbols:
 "_ftghbn_", referenced from:
     ___fitstools_MOD_read_fits_cut4.clone.2 in libhealpix.a(fitstools.o)
     ___fitstools_MOD_getsize_fits.clone.1 in libhealpix.a(fitstools.o)
     ___fitstools_MOD_getsize_fits in libhealpix.a(fitstools.o)
  ...
ld: symbol(s) not found
collect2: ld returned 1 exit status
```

and occurs when the CFITSIO installation script could not find a valid fortran compiler. To solve this problem

1. Go into the CFITSIO directory.
   Assuming that **ifort** is available on your system (it can be replaced below by **gfortran**, **g95**, **f77**, **f2c**, ...) type:
   ```
   ./configure FC=ifort
   make
   make install            (optional).
   ```

2. Then go back into the **HEALPix** directory and do
   ```
   ./configure             (making sure that you are using the newly created
   libcfitsio.a library)
   make
   make test
   ```

See also the note below on 64 bit architectures.

6.2.2   CFITSIO problems on systems with 64 bit architecture

1. **Linux, Mac OS X**

   If the **HEALPix** codes are compiled in 64 bits, and the GNU C Compiler (`gcc`) is used to compiled CFITSIO, then issue the following commands in the CFITSIO directory:

   ```
   ./configure FC='gcc -m64'
   make
   ```

   You can then force compilation to the same binary format by entering `-m64` when asked for the optimisation options in the **HEALPix** configure script.

2. **IRIX64**

   On a 64-bit architecture such as IRIX64, CFITSIO will have to be compiled in the same binary format as the **HEALPix** codes. This can be achieved by typing the following on the command line in the CFITSIO directory:

```
rm config.cache
setenv CC 'cc -n32'
./configure
make
```

Alternatively you can replace the `-n32` with `-64`. You can then force compilation to the same binary format by entering either `-n32` or `-64` when asked for the optimisation options in the **HEALPix** configure script.

6.2.3   CFITSIO linking problems

A particular problem encountered with the CFITSIO Version 2.0 release relates to the inclusion of various libraries within the system release for a given machine. This led to some modifications to the Makefile to include the specific library links `-lm -lnsl -lsocket` on SunOS, but only `-lm` for IRIX64. If your OS is not completely supported by the distribution, you may find this as one source of errors. The CFITSIO developers recommend compilation of the `testprog` routine. Inspection of the libraries linked after executing the `make testprog` statement will reveal those you need to include in the Makefile.

6.2.4   CFTISIO and Debian/Linux

Some problems have been reported on Debian/Linux systems during the linking to the CFITSIO library shipped with Linux. If these problems occur, try to recompile the CFITSIO library from scratch before linking to **HEALPix** .

6.3   `diff` shows that the test files are different from the supplied files

This by itself is no cause for concern. When comparing using a `diff` on the test files will most likely report a difference even when the installation has been successful. This may be due to the fact that different installations have different floating point representations. Also, the FITS files carry date information.

6.4   MIPSPro Compilers on SGI machines

Regrettably, the MIPSpro Compiler Version 7.20 has a compiler bug which cause run-time memory faults. We have not found any problems with Version 7.2.1.1m.

6.5 Try `unlimit`

If you have unforeseen problems at runtime, try `unlimit` (under csh or tcsh) or `ulimit` (under sh or bash), in order to increase the heap and stack memory size. It sometimes helps.

6.6 `hidl` usage

We have found that in very rare cases the alias `hidl` is not recognised by the user's system. Usually, this is related to the local system's IDL script. A quick-fix is achieved by setting the environment variable `IDL_STARTUP` to be equal to the **HEALPix** startup file `HEALPix_startup` **including** the directory path to the file. This enables the user to access the **HEALPix** IDL procedures simply by invoking IDL. For example, in the typical installation documented above for a user running the tcsh shell, the command
`setenv IDL_STARTUP /disk1/user1/HEALPix_2.15a/src/idl/HEALPix_startup`
should be issued (or added to the user's shell profile).

If the user already has an IDL startup file, then this should be merged with `HEALPix_startup`. This temporary solution does mean that the **HEALPix** IDL procedures are available in the `IDL_PATH` at all times, which may lead to conflicts with user-defined procedures. The `hidl` invocation was intended to circumvent these issues, allowing **HEALPix** IDL procedures to be available only when desired.

A proper fix requires the user to ask the local system administrator to adjust the local IDL script.

6.7 Mac OS X, X11 and IDL cursor

If the IDL cursor does not work correctly on X11 windows under Mac OS X, and the 2nd and 3rd button clicks are ineffective, type

- under Tiger (10.4.*):
  `defaults write com.apple.x11 wm_click_through -bool true`

- under Leopard (10.5.*) and Snow Leopard (10.6.*):
  `defaults write org.x.x11 wm_click_through -bool true`

at your X11 prompt and restart X11. (See also `mollcusor` documentation in "IDL Facilities").

6.8 Using GDL instead of IDL

GNU Data Language (GDL), is a *free* clone of IDL 6.0 (for more information see http://gnudatalanguage.sourceforge.net). Both the source code and precompiled executables for various platforms are available.

When used to run IDL-Healpix routines, GDL 0.9rc3 or more gives satisfactory results[2]. The calculations agree with those done under IDL, with comparable computation times, but a few features are missing in the production of Postscript, GIF and PNG files, as described below.

6.8.1   Specific requirements

HEALPix requires a few IDL routines that are not yet part of GDL. Among those are

- `congrid.pro`,

- `dl_dos.pro`[†],

- `dl_unix.pro`[†],

- `doc_library.pro`[†],

- `file_which.pro`,

- `interpol.pro`.

Most of these routines can be downloaded from http://idlastro.gsfc.nasa.gov/idl-libsrch.html. However, those marked with a [†] above can only be found in IDL packages (`$IDL_DIR/lib/*.pro`). They are required for online documentation of the **HEALPix** routines (ie `/HELP` keyword available on most routines).

Please note that GDL must be linked with `ImageMagick` during installation to produce PNG output files.

6.8.2   Impact of GDL limitations on HEALPix

- `Ximview` won't work under GDL 0.9rc3

- Because of problems with GDL's `point_lun` routine, the reading of FITS files with most of Astronomy User's Library routines won't be possible.

- Currently, the `cartview`, `gnomview`, `mollview` and `orthview` routines won't produce Postscript nor GIF outputs when run under GDL (but PNG files are OK, see above).

- In those same routines, the `TRANSPARENT` keyword will be ignored in the production of PNG files under GDL. For the same reasons, `hpx2gs` won't mark missing pixels as transparent in the output PNG file.

- The `healpix_doc` routine won't work under GDL.

---

[2]All the caveats listed below have been noticed in GDL v0.9rc3 and may be solved in subsequent versions. Please send all your questions *on* GDL directly to GDL developpers.

7 Appendix I: Recent Changes and New Features

7.1 Bug corrections and Improvements in Versions 2.15 and 2.15a

7.1.1 Fortran 90

- `remove_dipole`: removed confusing warning messages about unused masks and weights.

7.1.2 IDL

- `cartview, gnomview, mollview, orthview`:

  - export of projected map into a FITS file (`FITS` keyword), or an IDL array (`MAP_OUT` option) now available with all viewing routines,

  - added `CHARTHICK` support; accept array of `OUTLINE` structures (if they have the same fields),

  - correction of a bug (in `loaddata_healpix`) that was affecting the behavior of these viewing routines after consecutive calls with very partial cut-sky *and then* full-sky data sets [2.15a];

- `remove_dipole` now outputs the monopole and dipole covariance matrix;

- `write_fits_map, write_tqu, write_fits_sb`: `BAD_DATA` keyword added to FITS header;

- update of `astron` library routines (24-May-2010) for improved WCS support.

7.2 Bug corrections and Improvements in Version 2.14a

7.2.1 Fortran90

- correction of a numerical bug in `alm2map_der` routine that was affecting the accuracy of the Stokes parameter derivatives $\partial X/\partial\theta$, $\partial^2 X/(\partial\theta\partial\phi\sin\theta)$, $\partial^2 X/\partial\theta^2$, for $X = Q, U$ produced by `synfast` (bug detected by Wen Zhao, Cardiff University). See "Fortran Facilities" Appendix for details.

7.2.2 IDL

- `cartview, gnomview, mollview, orthview`:

- OUTLINE=, GRATICULE=, IGRATICULE= work again with virtual windows (WINDOW< 0)

- YPOS= and RETAIN= keywords active again

- PS= keyword fixed

- orthview: fixed problems with /SHADE keyword, which now outputs 8-byte (instead of 16-byte) PNG files

- ianafast, ismoothing: fixed problem with processing of polarized maps stored in memory.

- ud_grade: improved handling of flagged pixels on Double Precision input maps

- remove_dipole: COORD_IN= and COORD_OUT= now accept lower case values; /SILENT keyword added.

7.2.3   Java

- 64 bit java-**HEALPix** , supports Nside up to $= 2^{29} = 536870912$

  - now fully implemented

  - much faster pixel queries (eg query_disc)

- jhealpixSmall.jar: new smaller jar containing only main classes to be used into other applications or from the web (applets, ...)

- Java3d: upgrade/degrade fixed; color bar update fixed

- many others minor issues and javadocs fixed

7.3   Bug corrections and Improvements in Version 2.13a

7.3.1   general

- the configure script and its subroutines are now POSIX compliant (in particular, they should now run smoothly under Ubuntu Linux)

7.3.2   Fortran90

- new get_healpix_data_dir, get_healpix_main_dir, get_healpix_test_dir functions returning full path to **HEALPix** directories

7.3.3 IDL

- new `healpix_doc` routine to browse HTML and PDF documentations

- `cartview, gnomview, mollview, orthview`:

  - introduction of the `TRUECOLORS=` keyword to generate color image from 3 channel map

  - extended capability of the `TRANSPARENT=` keyword

  - addition of `MAP_OUT=` to `gnomview`

- improved compatibility with GDL (free IDL clone). See §6.8 for current GDL limitations.

- update of the `IDL-astron` library routines, which now require IDL 6.1 or more

- `fits2alm`: new `LMAX=` and `LMIN=` keywords

- `fits2cl`: new `LLFACTOR=` keyword

- `init_healpix` defines substructure with complete path to **HEALPix** subdirectories (test, data, bin)

- slightly faster `write_fits_cut4` and `write_fits_sb` routines.

- `ianafast, ismoothing`: solved problem with `W8DIR=` keyword.

7.4 Bug corrections and Improvements in Versions 2.12 and 2.12a

7.4.1 general

- the `Makefiles` are can now handle parallel compilation (ie `make -j` or `make --jobs`).

7.4.2 C++

- a warning (instead of an error) is issued in case of version mismatch between CFITSIO header and run time library

7.4.3 Fortran90

- to avoid confusion with a widespread library, the Healpix specific GIF routines library is now named `libhpxgif.a` instead of `libgif.a`. However, to avoid disrupting existing user-developed Makefiles, the library will remain available under both names for the time being.

- solved problem with `gfortran` and `g95` occuring over a line longer than 132 characters [2.12a]

7.4.4 IDL

- `ianafast`, `ismoothing`, `isynfast`: the `TMPDIR` keyword now works properly, and `$IDL_TMPDIR` is used as the default temporary directory.

- `ud_grade`: correctly flags bad output pixels with `bad_data` value when upgrading maps; now works for Nside > 8192 on cut sky data

- `cartview`, `gnomview`, `mollview`, `orthview`: using a virtual window (ie, `WINDOW`< 0) now allows faster generation of GIF and PNG files (especially useful over remote connections); added `RETAIN=` keyword; deals correctly with user provided `MIN` and `MAX` keywords in `/LOG` mode

- `orthview`: addition of `/SHADED` keyword for 3D rendering

- `ximview`: version 0.6.2, fixed bug in pixel coordinates [2.12a]

7.4.5 Java

- `getNSide` method in `HealpixIndex` class changed to `calculateNSide`.

- Area methods static - upgrade/degrade methods added.

8 Appendix II: Older changes

8.1 Bug corrections and Improvements in Version 2.11c

8.1.1 general

- `hpxconfig_functions.sh`: corrected typos, correctly detects recent `gfortran`;
- `configure`: added explanatory header

8.1.2 C

- edited `src/C/subs/Makefile` to fix potential problem with `test_chealpix2` code compilation

8.1.3 C++

- in **src/cxx/alice/planck.make**, replaced confusing `test` with `test_alice`

8.1.4 Fortran90

- `sky_ng_sim_bin` code now working
- use pointers in `map2alm_iterative` routine in order to fix unexplained crashes
- edited `create_alm` routine to work around a bug in with some versions of PGF90
- `write_minimal_header` corrected typo in FITS header for derivative maps

8.1.5 IDL

- fixed bug in `selectread`
- introduced **src/idl/misc/init_astrolib.pro**: wrapper to `astrolib`
- removed invocation to `!debug` in 11 FITS related routines

8.2 Bug corrections and Improvements in Version 2.11

8.2.1 general

- installation:
  - Fixed IDL installation problem under (ba)sh
  - Check consistency of cfitsio library with F90 routines
  - introduced command line options for `./configure`
  - fixed infinite loop in main Makefile's tidy and distclean options
- documentation:
  - "IDL routines": rightfully says that IDL6.0 or more recent is required

8.2.2 C

- Makefile no longer assumes that current directory ('.') is in PATH

8.2.3 C++

- fixed error affecting equatorial pixels in maps generated by `alm2map*`

8.2.4 Fortran90

- fixed problem with `TTYPE*` when writing FITS header (was also affecting IDL's `isynfast`)
- fixed crash causing bug in `map2alm_iterative` subroutine (was affecting `anafast, sky_ng_sims` and `smoothing` facilities)
- fixed un-initialized variables in `compute_statistics` subroutine
- fixed parameters parsing problem in interactive mode of `synfast, anafast, ...`

8.2.5   IDL

- issues warning when non-integer pixel indexes are fed to `nest2ring`, `ring2nest`, `pix2ang_*`, `pix2vec_*`, ...

- more stable and elegant behaviour of `isynfast` and `ianafast`

- `cartview`, `gnomview`, `mollview`, `orthview`:

   - deal properly with `MIN` and `MAX` in `ASINH` mode
   - polarization norm map can be offset (`POLARIZATION=1` mode)
   - original color table and plot settings are restored when leaving these routines

- `ximview`:

   - fixed problem with cut-sky FITS files
   - color scale bar added to PNG output

- cosmetic editions to `remove_dipole`

8.2.6   Java

8.3   Changes and New Features in Version 2.10

8.3.1   general

- single configure script and Makefile for C, C++, Fortran and IDL codes and libraries

8.3.2   C

- shared `chealpix` library available

8.3.3   C++

- new `alice` visualization facility for polarized maps

- new class Healpix_Base2 supporting Nside up to 2^{29}

- minor bug fixes

8.3.4   Fortran90

- The **free** Fortran90/95 compilers supported include:

   - Intel Fortran Compiler for linux based computers (version 8.1, 9.* or 10.*)
     http://www.intel.com/software/products/compilers/downloads/forlin.htm

- GNU Fortran 95 compiler (gfortran) included in GNU Compiler Collection *GCC* version 4.0.0 and up and available for Linux, Mac OSX, Windows, Sun ... platforms
  http://www.gnu.org/software/gcc/fortran/.
  GFortran binaries for all platforms can also be downloaded from
  http://gcc.gnu.org/wiki/GFortranBinaries.
  Please note that only the most recent versions of gfortran (Aug 2005 and later) compile HEALPix correctly, and v4.2.1 has given satisfying results so far, including native OpenMP support.
- g95 compiler available for Linux, Mac OSX, Windows, Sun and HP platforms
  http://g95.sf.net

- New facilities

  - The anafast facility can now compute the cross-correlations of two different maps.
  - The sky_ng_sim facility (Rocha et al, 2005), to produce non-Gaussian CMB temperature maps, has been added.

- New routines:

  - alm2map_spin: synthesis of maps of arbitrary spin
  - healpix_modules: meta-module
  - map2alm_iterative: iterative analysis of map
  - map2alm_spin: analysis of maps of arbitrary spin
  - write_minimal_header: routine to write minimal FITS header
  - parse_check_unused: prints out parameters present in parameter file but not used by the code.

- Improved routines:

  - query_strip: the `inclusive` option now returns *all* (and only) the pixels overlapping, even partially, with the strip
  - query_disc: when the disc center is on one of the poles, *only* the pixels overlapping with the disc are now returned.
  - remove_dipole: can now deal with non-uniform pixel weights.
  - parse_init: silent mode
  - parse_string: can expand environment variables (${XXX}) and leading ~/

8.3.5 IDL

- New features

  - Using 64 bit integers available since version 5.2 of IDL the maximum resolution parameter Nside supported has increased from $2^{13} = 8192$ to $2^{29} = 536870912$, corresponding to $3.46\,10^{18}$ pixels on the sphere.
  - FITS files larger than 2GB can now be read/written on systems supporting 64 bit addressing.

- New routines include

  - `ximview`: visualisation routine developed by J. P. Leahy intended for quick-look inspection of HEALPix images (as well as ordinary 2-D images) at the level of individual pixels. Features include panning, zooming, blinking, image statistics and peak finding.

- hpx2gs: turns a healpix data set into a Google Earth/Google Sky-compatible image

- ianafast: interface to (F90) anafast and (C++) anafast_cxx facilities

- isynfast: interface to F90 synfast facility

- ismoothing: interface to F90 smoothing facility

- bin_llcl: $C(l)$ binning

- bl2fits: writes $B(l)$ or $W(l)$ window into FITS file

- neighbours_nest, neighbours_ring: find immediate neighbours of a given pixel

- query_strip: find pixels lying within a colatitude strip

- Routines with extended/improved user interface or new functionalities include

  - mollview, gnomview, cartview, orthview:

    * ONLINE keyword is now redundant,
    * introduction of GLSIZE and IGLSIZE to control automatic labeling of graticules,
    * addition of SILENT and EXECUTE keywords,
    * addition of ASINH keyword to allow better visualisation of highly contrasted maps.
    * under certain circumstances, can process high resolution cut sky data sets without creating full sky dummy maps,
    * accept gzip compressed FITS files,
    * accept polarized cut sky maps,
    * accept multi-dimensional online arrays,
    * more robust OUTLINE option.

  - median_filter: bugs correction

  - ud_grade: more robust user interface

  - change_polcconv: new /FORCE keyword

  - remove_dipole: more accurate

  - query_disc: when the disc center is on one of the poles, *only* the pixels overlapping with the disc are now returned.

- Miscellaneous

  - mollcursor, gnomcursor...: an X11 patch is given so that these routines work under Mac OS X 10.4 and 10.5.

8.3.6   Java

- Almost all query pixels related methods ported to Java have been added like:

  - query_triangle, query_polygon, etc,

  - a tool class for operating on healpix map.

- OO Data model added:

  - HealpixMap data model interface introduced,

  - AbstractHealpixMap interface to deal with 3d viewer visualization.

- Java graphical user interface included to plot healpix maps object into a 3d sphere using java3d api:

  - 3 button Mouse operating 3d pixelised sphere: zoom, rotation and translation,
  - tool tip display value and angular position of the pixel selected,
  - color bar, grid, axes,
  - transparencies selector,
  - nside number selector,
  - face number selector.

- Bugs fixed like:

  - crosses the 0/360 line give correct query triangle solution,
  - library uses one constants class to define used constants,
  - SpatialVector extends the vector class from vecmath,
  - Change all float to double,
  - Many other minor bug fixed.

8.4   Changes and New Features in Version 2.00

- Free / Open source software license

- New web page: http://healpix.jpl.nasa.gov

- HTML and PDF documentations are available in `[HEALPix]/doc/html/main.htm` and `[HEALPix]/doc/pdf/`

- C++ implementation of almost all codes and routines

- Fortran90:

  - 2 to 3 times faster Spherical Harmonics tools (ie, synfast, anafast, smoothing, plmgen)
  - parallelization for distributed memory architectures
  - More **free** Fortran90/95 compilers supported:
    * Intel Fortran Compiler for linux based computers (version 8.1 or 9.0) http://www.intel.com/software/products/compilers/downloads/forlin.htm
    * GNU Fortran 95 compiler (gfortran) included in GNU Compiler Collection *GCC* version 4.0.0 and up and available for Linux, Mac OSX, Windows, Sun ... platforms http://www.gnu.org/software/gcc/fortran/ Please note that only the most recent CVS versions of gfortran (Aug 2005 and later) compile HEALPix correctly!
    * g95 compiler available for Linux, Mac OSX, Windows, Sun and HP platforms http://g95.sf.net

- C++ and F90 codes

  - (almost) identical user interface
  - parallelization for shared memory architectures
  - *new facilities:* median filtering of **HEALPix** map and alteration/rotation of Spherical Harmonics coefficients

 – single and double precision I/O

 – speed optimization of FITS I/O

 – more platforms/compilers supported (including Mac OS X, 32/64bit architecture, etc.)

 – *new, longer period, random number generator*

 – *linkage to FFTW has been discontinued*

 – precomputed ring weights used for quadrature of the Spherical Harmonics are now available for all resolution parameters $N_{\text{side}} \leq 8192$

- Java implementation of pixel tools

- IDL:

 – median filtering facility

 – speed optimization of FITS I/O

 – more options in file reading and visualisation routines

8.5   Changes and New Features in Version 1.20

- An extensive HTML documentation is included in the package (in `[HEALPix]/doc/html/main.htm`). It is also available at the web page `http://healpix.jpl.nasa.gov`, as well as the PostScript documention.

- The convention used for the normalisation of the polarization power spectra has been changed to match that of CMBFAST. This subject and its implications are described in details in the **HEALPix** Primer

- The pixel window functions are now provided for both temperature and polarization. Recalculation of the temperature window for $N_{side} > 128$ induce a slight change of these functions (see the "**HEALPix** Primer" for details).

- Because of its obsolete F66 features that limited its portability with current f90/f95 compilers, the FFTPACK module used in version 1.10 is no longer in use. **HEALPix** 1.2 can either use a different self contained FFT module shipped with the package, or use the freely available **FFTW** library (see section 2). The script file asks which FFT the user wants to use. If selected, FFTW must be correctly installed (in double precision) before installing Healpix.

- It is now possible to use an OpenMP parallelisation of the Spherical Harmonics Transforms (see § 2).

- The compilers now supported under **Linux** systems are

 – Intel Fortran Compiler, available for **free** `http://www.intel.com/software/products/compilers/downloads/forlin.htm` (tested for version 6.01 and 7.0)

 – Lahey/Fujitsu Compiler, both **free** trial and commercial versions `http://www.lahey.com/linux.htm`

 – NAGWare f95, commercial `http://www.nag.co.uk/nagware/NP.asp`

 – Portland Group Compiler, commercial `http://www.pgroup.com/`

 – Fujitsu Compiler, commercial

- The **Mac** Operating System **Darwin** is now supported. To this date (Jan 2003) it has only been tested with NAGWare f95 http://www.nag.co.uk/nagware/NP.asp.

- A small subset of routines for the pixel to sky coordinate conversion is now available in C (see the documentation "C Subroutines Overview").

- New Fortran subroutines have been added or upgraded (see the documentation "Fortran90 Subroutines Overview") and most of the Fortran facilities have been upgraded (see the documentation "Fortran90 Facilities User Guidelines")

- several Fortran routines have been renamed and merged.

  - The module `wrap_fits` has been renamed `head_fits` to reflect its extended capabilities,

  - some routines no longer in use have been moved to `obsolete` module

- New IDL routines have been added, and several routines have been renamed or merged. See the documentation "IDL Facilities Overview"

- The configure script automatically creates a (system specific) profile defining useful environment variables and aliases. *In particular, the system variable* `HEALPIX` *must be correctly defined (as the full path to the **HEALPix** directory) prior to running the pipeline.*

- All legacy f77 or f66 codes have been removed or replaced in order to improve the portability

- The script file has been improved to, among other things, get rid of problems appearing under SunOS

8.6   Changes and New Features in Version 1.10

- Only native `make` is required for building the facilities, although we do still recommend GNU make.

- The configure script automatically creates a (system specific) profile defining useful environment variables and aliases

- An IDL startup file is provided to ease use of IDL tools. Users should invoke IDL using the command `hidl` to take advantage of these features

- Various bug fixes and code modifications have been completed to improve compatibility and usability of software with NAG F90 and F95 compilers, and the COMPAQ (DEC) Alpha compiler

- New Fortran subroutines and functions include:
  `ang2vec, vec2ang, npix2nside, nside2npix`

- New IDL routines include:
  `add_nside_fits.pro, add_units_fits.pro, gaussbeam.pro, getdisc_ring.pro,`
  `healpixwindow.pro, npix2nside.pro, nside2npix.pro, remove_dipole.pro,`
  `today_fits.pro, ud_grade.pro`

- Those routines from The IDL Astronomy User's Library and the COBE (IDL) Analysis Software library required for use by the **HEALPix** IDL facilities are bundled with the distribution